

DDDDDDDDDDDDDD	CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD	CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD	CCCCCCCCCCCC	LLL
DDD	DDD CCC	LLL
DDDDDDDDDDDDDD	CCCCCCCCCCCC	LLLLLLLLLLLL
DDDDDDDDDDDDDD	CCCCCCCCCCCC	LLLLLLLLLLLL
DDDDDDDDDDDDDD	CCCCCCCCCCCC	LLLLLLLLLLLL

FILE ID CONVERT

E 10

(2)	60	DECLARATIONS
(3)	78	ASCII NUMERIC VALUE CONVERSION ROUTINE
(4)	150	CONVERT BINARY TO 8-DIGIT HEX
(5)	197	CONVERT BINARY TO ZERO SUPPRESSED ASCII
(6)	250	EDIT THE STRING

0000 1 .TITLE CONVERT - DCL VALUE CONVERSION AND EDITING SUBROUTINES
0000 2 :.IDENT 'V04-000'
0000 3 :
0000 4 :
0000 5 :*****
0000 6 :
0000 7 : COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 : DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 : ALL RIGHTS RESERVED.
0000 10 :
0000 11 : THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 : ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 : INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 : COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 : OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 : TRANSFERRED.
0000 17 :
0000 18 : THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 : AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 : CORPORATION.
0000 21 :
0000 22 : DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 : SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :
0000 25 :
0000 26 :*****
0000 27 :
0000 28 :
0000 29 :++
0000 30 :FACILITY: DCL
0000 31 :
0000 32 :ABSTRACT: Value conversion and editing subroutines
0000 33 :
0000 34 :AUTHOR: Peter George
0000 35 :
0000 36 :DATE: 01-MAR-1983
0000 37 :
0000 38 :MODIFIED BY:
0000 39 :
0000 40 :V03-006 HWS0096 Harold Schultz 27-Jul-1984
0000 41 : Fix HWS0082 to correctly handle null strings.
0000 42 :
0000 43 :V03-005 HWS0082 Harold Schultz 19-Jul-1984
0000 44 : Fix DCL\$CNVASCBIN to not return success if processing
0000 45 : a string of all unary operators without any numbers.
0000 46 :
0000 47 :V03-004 PCG0004 Peter George 11-Jan-1984
0000 48 : Add lowercase to edit.
0000 49 :
0000 50 :V03-003 PCG0003 Peter George 15-Jul-1983
0000 51 : Use multi-national upcase algoritm.
0000 52 :
0000 53 :V03-002 PCG0002 Peter George 15-Jun-1983
0000 54 : Fix bug in DCL\$CBTA_HEXSTR.
0000 55 :
0000 56 :V03-001 PCG0001 Peter George 09-May-1983
0000 57 : Add DCL\$CBTA_HEXSTR.
0000 58 :
0000 59 :
0000 60 :
0000 61 :
0000 62 :
0000 63 :
0000 64 :
0000 65 :
0000 66 :
0000 67 :
0000 68 :
0000 69 :
0000 70 :
0000 71 :
0000 72 :
0000 73 :
0000 74 :
0000 75 :
0000 76 :
0000 77 :
0000 78 :
0000 79 :
0000 80 :
0000 81 :
0000 82 :
0000 83 :
0000 84 :
0000 85 :
0000 86 :
0000 87 :
0000 88 :
0000 89 :
0000 90 :
0000 91 :
0000 92 :
0000 93 :
0000 94 :
0000 95 :
0000 96 :
0000 97 :
0000 98 :
0000 99 :
0000 100 :
0000 101 :
0000 102 :
0000 103 :
0000 104 :
0000 105 :
0000 106 :
0000 107 :
0000 108 :
0000 109 :
0000 110 :
0000 111 :
0000 112 :
0000 113 :
0000 114 :
0000 115 :
0000 116 :
0000 117 :
0000 118 :
0000 119 :
0000 120 :
0000 121 :
0000 122 :
0000 123 :
0000 124 :
0000 125 :
0000 126 :
0000 127 :
0000 128 :
0000 129 :
0000 130 :
0000 131 :
0000 132 :
0000 133 :
0000 134 :
0000 135 :
0000 136 :
0000 137 :
0000 138 :
0000 139 :
0000 140 :
0000 141 :
0000 142 :
0000 143 :
0000 144 :
0000 145 :
0000 146 :
0000 147 :
0000 148 :
0000 149 :
0000 150 :
0000 151 :
0000 152 :
0000 153 :
0000 154 :
0000 155 :
0000 156 :
0000 157 :
0000 158 :
0000 159 :
0000 160 :
0000 161 :
0000 162 :
0000 163 :
0000 164 :
0000 165 :
0000 166 :
0000 167 :
0000 168 :
0000 169 :
0000 170 :
0000 171 :
0000 172 :
0000 173 :
0000 174 :
0000 175 :
0000 176 :
0000 177 :
0000 178 :
0000 179 :
0000 180 :
0000 181 :
0000 182 :
0000 183 :
0000 184 :
0000 185 :
0000 186 :
0000 187 :
0000 188 :
0000 189 :
0000 190 :
0000 191 :
0000 192 :
0000 193 :
0000 194 :
0000 195 :
0000 196 :
0000 197 :
0000 198 :
0000 199 :
0000 200 :
0000 201 :
0000 202 :
0000 203 :
0000 204 :
0000 205 :
0000 206 :
0000 207 :
0000 208 :
0000 209 :
0000 210 :
0000 211 :
0000 212 :
0000 213 :
0000 214 :
0000 215 :
0000 216 :
0000 217 :
0000 218 :
0000 219 :
0000 220 :
0000 221 :
0000 222 :
0000 223 :
0000 224 :
0000 225 :
0000 226 :
0000 227 :
0000 228 :
0000 229 :
0000 230 :
0000 231 :
0000 232 :
0000 233 :
0000 234 :
0000 235 :
0000 236 :
0000 237 :
0000 238 :
0000 239 :
0000 240 :
0000 241 :
0000 242 :
0000 243 :
0000 244 :
0000 245 :
0000 246 :
0000 247 :
0000 248 :
0000 249 :
0000 250 :
0000 251 :
0000 252 :
0000 253 :
0000 254 :
0000 255 :
0000 256 :
0000 257 :
0000 258 :
0000 259 :
0000 260 :
0000 261 :
0000 262 :
0000 263 :
0000 264 :
0000 265 :
0000 266 :
0000 267 :
0000 268 :
0000 269 :
0000 270 :
0000 271 :
0000 272 :
0000 273 :
0000 274 :
0000 275 :
0000 276 :
0000 277 :
0000 278 :
0000 279 :
0000 280 :
0000 281 :
0000 282 :
0000 283 :
0000 284 :
0000 285 :
0000 286 :
0000 287 :
0000 288 :
0000 289 :
0000 290 :
0000 291 :
0000 292 :
0000 293 :
0000 294 :
0000 295 :
0000 296 :
0000 297 :
0000 298 :
0000 299 :
0000 300 :
0000 301 :
0000 302 :
0000 303 :
0000 304 :
0000 305 :
0000 306 :
0000 307 :
0000 308 :
0000 309 :
0000 310 :
0000 311 :
0000 312 :
0000 313 :
0000 314 :
0000 315 :
0000 316 :
0000 317 :
0000 318 :
0000 319 :
0000 320 :
0000 321 :
0000 322 :
0000 323 :
0000 324 :
0000 325 :
0000 326 :
0000 327 :
0000 328 :
0000 329 :
0000 330 :
0000 331 :
0000 332 :
0000 333 :
0000 334 :
0000 335 :
0000 336 :
0000 337 :
0000 338 :
0000 339 :
0000 340 :
0000 341 :
0000 342 :
0000 343 :
0000 344 :
0000 345 :
0000 346 :
0000 347 :
0000 348 :
0000 349 :
0000 350 :
0000 351 :
0000 352 :
0000 353 :
0000 354 :
0000 355 :
0000 356 :
0000 357 :
0000 358 :
0000 359 :
0000 360 :
0000 361 :
0000 362 :
0000 363 :
0000 364 :
0000 365 :
0000 366 :
0000 367 :
0000 368 :
0000 369 :
0000 370 :
0000 371 :
0000 372 :
0000 373 :
0000 374 :
0000 375 :
0000 376 :
0000 377 :
0000 378 :
0000 379 :
0000 380 :
0000 381 :
0000 382 :
0000 383 :
0000 384 :
0000 385 :
0000 386 :
0000 387 :
0000 388 :
0000 389 :
0000 390 :
0000 391 :
0000 392 :
0000 393 :
0000 394 :
0000 395 :
0000 396 :
0000 397 :
0000 398 :
0000 399 :
0000 400 :
0000 401 :
0000 402 :
0000 403 :
0000 404 :
0000 405 :
0000 406 :
0000 407 :
0000 408 :
0000 409 :
0000 410 :
0000 411 :
0000 412 :
0000 413 :
0000 414 :
0000 415 :
0000 416 :
0000 417 :
0000 418 :
0000 419 :
0000 420 :
0000 421 :
0000 422 :
0000 423 :
0000 424 :
0000 425 :
0000 426 :
0000 427 :
0000 428 :
0000 429 :
0000 430 :
0000 431 :
0000 432 :
0000 433 :
0000 434 :
0000 435 :
0000 436 :
0000 437 :
0000 438 :
0000 439 :
0000 440 :
0000 441 :
0000 442 :
0000 443 :
0000 444 :
0000 445 :
0000 446 :
0000 447 :
0000 448 :
0000 449 :
0000 450 :
0000 451 :
0000 452 :
0000 453 :
0000 454 :
0000 455 :
0000 456 :
0000 457 :
0000 458 :
0000 459 :
0000 460 :
0000 461 :
0000 462 :
0000 463 :
0000 464 :
0000 465 :
0000 466 :
0000 467 :
0000 468 :
0000 469 :
0000 470 :
0000 471 :
0000 472 :
0000 473 :
0000 474 :
0000 475 :
0000 476 :
0000 477 :
0000 478 :
0000 479 :
0000 480 :
0000 481 :
0000 482 :
0000 483 :
0000 484 :
0000 485 :
0000 486 :
0000 487 :
0000 488 :
0000 489 :
0000 490 :
0000 491 :
0000 492 :
0000 493 :
0000 494 :
0000 495 :
0000 496 :
0000 497 :
0000 498 :
0000 499 :
0000 500 :
0000 501 :
0000 502 :
0000 503 :
0000 504 :
0000 505 :
0000 506 :
0000 507 :
0000 508 :
0000 509 :
0000 510 :
0000 511 :
0000 512 :
0000 513 :
0000 514 :
0000 515 :
0000 516 :
0000 517 :
0000 518 :
0000 519 :
0000 520 :
0000 521 :
0000 522 :
0000 523 :
0000 524 :
0000 525 :
0000 526 :
0000 527 :
0000 528 :
0000 529 :
0000 530 :
0000 531 :
0000 532 :
0000 533 :
0000 534 :
0000 535 :
0000 536 :
0000 537 :
0000 538 :
0000 539 :
0000 540 :
0000 541 :
0000 542 :
0000 543 :
0000 544 :
0000 545 :
0000 546 :
0000 547 :
0000 548 :
0000 549 :
0000 550 :
0000 551 :
0000 552 :
0000 553 :
0000 554 :
0000 555 :
0000 556 :
0000 557 :
0000 558 :
0000 559 :
0000 560 :
0000 561 :
0000 562 :
0000 563 :
0000 564 :
0000 565 :
0000 566 :
0000 567 :
0000 568 :
0000 569 :
0000 570 :
0000 571 :
0000 572 :
0000 573 :
0000 574 :
0000 575 :
0000 576 :
0000 577 :
0000 578 :
0000 579 :
0000 580 :
0000 581 :
0000 582 :
0000 583 :
0000 584 :
0000 585 :
0000 586 :
0000 587 :
0000 588 :
0000 589 :
0000 590 :
0000 591 :
0000 592 :
0000 593 :
0000 594 :
0000 595 :
0000 596 :
0000 597 :
0000 598 :
0000 599 :
0000 600 :
0000 6

CONVERT
V04-000

- DCL VALUE CONVERSION AND EDITING SUBRO ^{H 10} 15-SEP-1984 23:42:19 VAX/VMS Macro V04-00
4-SEP-1984 23:39:53 [DCL.SRC]CONVERT.MAR;1

Page 2
(1)

0000 58 ;--

0000	60	.SBttl DECLARATIONS	
0000	61	:	
0000	62	: MACRO LIBRARY CALLS	
0000	63	:	
0000	64	WRKDEF	: DEFINE COMMAND WORK AREA
0000	65	EDITDEF	: DEFINE EDIT FLAGS
0000	66	\$CLIMSGDEF	: CLI MESSAGE DEFINITIONS
0000	67		
00000000	68	.PSECT DCLS\$ZCODE	BYTE, RD, NOWRT
0000	69		
0000	70	:	
0000	71	: OWN STORAGE:	
0000	72	:	
58 44 4F	0000	73 CNVRDX: .ASCII \0DX\	: CONVERSION RADIX CONTROLS
08 0A 10	0003	74 RADIX: .BYTE 16,10,8	: CORRESPONDING RADIX VALUES
	0006	75 DIGIT_TABLE:	: HEX DIGITS
42 41 39 38 37 36 35 34 33 32 31 30	0006	76 .ASCII '0123456789ABCDEF'	
46 45 44 43	0012		

0016 78 .SBTTL ASCII NUMERIC VALUE CONVERSION ROUTINE
 0016 79 ++
 0016 80
 0016 81 DCL\$CNVASCBIN - CONVERT ASCII STRING TO BINARY VALUE
 0016 82
 0016 83 THIS ROUTINE IS CALLED TO CONVERT AN ASCII STRING TO A BINARY VALUE.
 0016 84
 0016 85 INPUTS:
 0016 86
 0016 87 R1 = DEFAULT RADIX INDICATOR, 0=HEX, 1=DECIMAL, 2=OCTAL
 0016 88 R2/R3 = QUADWORD DESCRIPTOR OF VALUE
 0016 89
 0016 90 OUTPUTS:
 0016 91
 0016 92 R0 = 0 IF SUCCESSFUL, NONZERO IF FAILURE
 0016 93 R1 = CONVERTED BINARY VALUE
 0016 94
 0016 95 --
 0016 96 DCL\$CNVASCBIN:: : CONVERT IN SPECIFIED RADIX
 51 51 DD 0016 97 PUSHL R1 : SAVE R1
 51 52 7D 0018 98 MOVQ R2,R1 : SET UP THE ARGUMENT REGISTERS
 0119 30 001B 99 BSBW DCL\$TRIM : TRIM AND UPCASE THE STRING
 52 51 7D 001E 100 MOVQ R1,R2 : RETURN STRING TO R2/R3
 51 8ED0 0021 101 POPL R1 : RESTORE R1
 0024 102
 0024 103 DCL\$CNVNOEDIT:: : CONVERT NUMERIC DECIMAL RADIX
 7E D4 0024 104 CLRL -(SP) : SAVE ROOM FOR 'NUMBER SEEN' FLAG
 6243 9F 0026 105 PUSHAB (R2)[R3] : SAVE ADDRESS OF END OF STRING
 00 DD 0029 106 PUSHL #0 : SET NO NEGATE FLAG
 50 52 DD 002B 107 MOVL R2,R0 : TEST FOR ZERO LENGTH STRING
 56 13 002E 108 BEQL 70\$: IF IT IS THE NULL STRING-RETURN ZERO
 08 AE 01 DD 0030 109 MOVL #1,8(SP) : INIT. 'NUMBER SEEN' FLAG
 63 25 91 0034 110 CMPB #^A/%,(R3) : RADIX CHANGE OPERATOR?
 0E 12 0037 111 BNEQ 10\$: BR IF NO
 CO AF 03 83 3A 0038 112 INCL R3 : SKIP OVER OPERATOR
 44 13 0040 113 LOCC (R3)+,#3,CNVRDX : FIND RADIX SPECIFIER
 51 70 9E 0042 114 BEQL 70\$: BR IF NONE RECOGNIZED
 ED 11 0045 115 MOVAB -(R0),R1 : SET RADIX INDICATOR
 63 28 91 0047 116 BRB 5\$: LOOK FOR AN OTHER SET
 07 13 004A 117 10\$: CMPB #^A/+/, (R3) : CHECK FOR UNIARY OPERATOR OF PLUS
 63 2D 91 004C 118 BEQL 20\$: BR IF YES
 06 12 004F 119 CMPB #^A/-/, (R3) : HOW ABOUT MINUS
 6E D6 0051 120 BNEQ 30\$: BR IF NO
 53 D6 0053 121 INCL (SP) : NEGATE THE NEGATIVE FLAG
 DD 11 0055 122 20\$: INCL R3 : SKIP THE UNIARY OPERATOR
 52 A8 AF 41 9A 0057 124 30\$: MOVZBL RADIX[R1],R2 : LOOK FOR ANOTHER
 50 7C 005C 125 CLRQ R0 : SET ACTUAL RADIX OF CONVERSION
 51 63 30 83 005E 126 40\$: SUBB3 #^A/0/, (R3),R1 : START WITH RESULT AND WORK VALUE OF 0
 22 19 0062 127 BLSS 70\$: GET NEXT NUMBER MINUS ASCII BIAS
 09 51 91 0064 128 CMPB R1, #9 : BR IF NOT A NUMERIC CHARACTER
 08 15 0067 129 BLEQ 60\$: WAS IT A NUMERIC DIGIT?
 11 51 91 0069 130 CMPB R1, #<^A/A/-^A/0/> : BR IF YES
 18 19 006C 131 BLSS 70\$: HEX RADIX CHARACTER?
 51 07 82 006E 132 SUBB #<<^A/A/-^A/0/>-10>,R1 : BR IF NO - TERMINATE THE CONVERT
 52 51 91 0071 133 60\$: CMPB R1, R2 : CONVERT TO BINARY VALUE
 10 18 0074 134 BGEQ 70\$: WITHIN THE RADIX?
 : BR IF OUT OF THE RANGE

50	51	50	52	7A	0076	135	EMUL	R2,R0,R1,R0	: FIND CURRENT TOTAL
		08	AE	D4	0078	136	CLRL	8(\$P)	: INDICATE AT LEAST ONE NUMBER PROCESSED
		53		D6	007E	137	INCL	R3	: POINT AT NEXT CHARACTER
	04	AE	53	D1	0080	138	CMPL	R3,4(SP)	: TIME TO QUIT?
				D8	12	0084	BNEQ	40\$: BR IF NO
		51	50	DO	0086	140	70\$:	MOVL	: SET RESULT
		03	8E	E9	0089	141	BLBC	(SP)+,80\$: TEST NEGATE FLAG
	50	8E	53	CE	008C	142	MNEGL	R1,R1	: NEGATE THE SOURCE
		03	8E	C3	008F	143	80\$:	SUBL	: SET NUMBER OF UN-PROCESSED BYTES
		50	01	DO	0093	144	BLBC	R3,(SP)+,90\$: IF A NUMBER SEEN, FLAG OK AS IS
					0096	145	MOVL	#1,R0	: IF NO NUMBER SEEN, UNCONDITIONALLY
					0099	146			SET TO FAILURE
		05			0099	147	90\$:	RSB	: BACK TO THE CALLER
					009A	148			

		009A	150	.SBTTL CONVERT BINARY TO 8-DIGIT HEX					
		009A	151	:+ DCL\$CBTA_HEX - CONVERT BINARY TO HEX STRING					
		009A	152	CONVERT A BINARY NUMBER TO A HEX STRING OF EXACTLY 8 DIGITS.					
		009A	153						
		009A	154						
		009A	155	INPUTS:					
		009A	156						
		009A	157	R0 = NUMBER TO BE CONVERTED.					
		009A	158						
		009A	159	OUTPUTS:					
		009A	160						
		009A	161	R1 = LENGTH OF CONVERTED VALUE.					
		009A	162	R2 = ADDRESS OF CONVERTED VALUE.					
		009A	163						
		009A	164	:-					
		009A	165						
		009A	166	DCL\$CBTA_HEX::					
51	04	DD	009A	167	MOVL	#4,R1	:CREATE DESCRIPTOR OF HEX STRING		
52	50	DD	009D	168	PUSHL	R0	:		
	5E	DD	009F	169	MOVL	SP,R2			
	03	10	00A2	170	BSBB	DCL\$CBTA_HEXSTR	:GET CONVERTED VALUE		
	8E	D5	00A4	171	TSTL	(SP)+	:RESTORE THE STACK		
			00A6	172	RSB		:RETURN		
			00A7	173					
			00A7	174	DCL\$CBTA_HEXSTR::				
04 AE	5E	51	C2	00A7	175	SUBL	R1,SP	:ALLOCATE SCRATCH BUFFER ON THE STACK	
		51	DD	00AA	176	PUSHL	R1	:SAVE INPUT LENGTH	
	62	51	28	00AC	177	MOVC	R1,(R2),4(SP)	:COPY INPUT BUFFER	
		55	8ED0	00B1	178	POPL	R5	:RESTORE INPUT LENGTH	
	52	F486	CA	00B4	179	MOVL	WRK_L_EXPANDPTR(R10),R2	:MARK POSITION IN OUTPUT BUFFER	
	53	55	01	78	00B9	ASHL	#1,R5,R3	:DOUBLE THE NUMBER OF INPUT BYTES	
			00BD	180					
			00BD	181					
50	6E45	04	55	D7	00BD	182			
	50	FF3C	CF40	EF	00BF	183	10\$: DECL	R5	:START WITH LAST BYTE OF DATA
			FF32	9A	00C5	184	EXTZV	#4,#4,(SP)[R5],R0	:GET SECOND NIBBLE OF BYTE
50	6E45	04	00	EF	00CE	185	MOVZBL	DIGIT_TABLE[R0],R0	:CONVERT DIGIT TO ASCII
	50	FF2D	CF40	9A	00D4	186	BSBW	DCL\$PUTCHAR	:PUT THE CHAR IN THE OUTPUT BUFFER
			FF23	30	00DA	187	EXTZV	#0,#4,(SP)[R5],R0	:GET FIRST NIBBLE OF BYTE
		DF	55	F4	00DD	188	MOVZBL	DIGIT_TABLE[R0],R0	:CONVERT DIGIT TO ASCII
					00E0	189	BSBW	DCL\$PUTCHAR	:PUT THE CHAR IN THE OUTPUT BUFFER
					00E0	190	S0BGEQ	R5,10\$:BRANCH WHILE MORE DIGITS
51	F486	CA	52	C3	00E0	191	20\$: SUBL3	R2,WRK_L_EXPANDPTR(R10),R1	:GET DESCRIPTOR OF RESULTANT STRING
50	F486	CA	52	DD	00E6	192	MOVL	R2,WRK_L_EXPANDPTR(R10)	:RESET EXPANSION BUFFER POINTER
	51	FF	8F	78	00EB	193	ASHL	#-1,R1,R0	:GET THE NUMBER OF INPUT BYTES
	5E	50	CO	00F0	194	ADDL	R0,SP	:POP SCRATCH BUFFER OFF THE STACK	
			00F3	195	RSB				

00F4 197 .SBTTL CONVERT BINARY TO ZERO SUPRESSED ASCII
 00F4 198 :+
 00F4 199 : CBTA_DEC - CONVERT BINARY TO ASCII BASE TEN
 00F4 200 : CBTA_OCT - CONVERT BINARY TO ASCII BASE EIGHT
 00F4 201 :
 00F4 202 : THESE ROUTINES ARE CALLED TO CONVERT A BINARY NUMBER TO A LEFT JUSTIFIED, ZERO
 00F4 203 : SUPRESSED, ASCII STRING. THE RESULTANT STRING IS PLACED IN THE EXPANSION
 00F4 204 : BUFFER AND THE EXPANSION POINTER IS UPDATED.
 00F4 205 :
 00F4 206 : INPUTS:
 00F4 207 :
 00F4 208 : R0 = NUMBER TO BE CONVERTED.
 00F4 209 :
 00F4 210 : OUTPUTS:
 00F4 211 :
 00F4 212 : R1 = LENGTH OF CONVERTED VALUE.
 00F4 213 : R2 = ADDRESS OF CONVERTED VALUE.
 00F4 214 :
 00F4 215 : .ENABL LSB
 00F4 216 :
 53 08 9A 00F4 217 DCLSCBTA OCT:: : CONVERT BINARY TO ASCII BASE EIGHT
 03 11 00F7 218 MOVZBL #8,R3 : SET CONVERSION RADIX
 52 F486 0A 9A 00F9 220 DCLSCBTA DEC:: : CONVERT BINARY TO ASCII BASE TEN
 CA D0 00FC 221 MOVZBL #10,R3 : SET CONVERSION RADIX
 50 D5 0101 222 10\$: MOVL WRK_L_EXPANDPTR(R10),R2 : MARK POSITION IN BUFFER
 OC 18 0103 223 TSTL R0 : IS NUMBER NEGATIVE?
 54 50 CE 0105 224 BGEQ 15\$: NO, THEN SKIP
 50 2D 9A 0108 225 MNEGL R0,R4 : YES, THEN CHANGE SIGN AND SAVE VALUE
 FEF2' 30 010B 226 MOVZBL #^A'-' R0 : GET NEGATIVE SIGN
 50 54 DO 010E 227 BSBW DCLSPUTCHAR : PUT IT IN THE EXPANSION BUFFER
 OC 10 0111 228 MOVL R4,R0 : RECOVER VALUE
 51 F486 CA 52 C3 0113 229 15\$: BSBB 20\$: PUT ASCII CHARACTERS INTO BUFFER
 F486 CA 52 DO 0119 230 SUBL3 R2,WRK_L_EXPANDPTR(R10),R1 : GET DESCRIPTOR OF RESULTANT STRING
 05 011E 231 MOVL R2,WRK_L_EXPANDPTR(R10) : RESET EXPANSION BUFFER POINTER
 011F 232 RSB :
 011F 233 :
 011F 234 : RECURSIVE ROUTINE TO OUTPUT THE ASCII NUMBER, HIGH ORDER DIGITS FIRST
 011F 235 : WITHOUT ANY LEADING SPACES OR ZEROS.
 011F 236 :
 51 50 51 D4 011F 237 20\$: CLR1 R1 : CLEAR HIGH PART OF DIVIDEND
 7E 50 53 78 0121 238 EDIV R3,R0,R0,R1 : ISOLATE NEXT DIGIT
 30 C1 0126 239 ADDL3 #^A'0',R1,-(SP) : CONVERT DIGIT TO ASCII AND SAVE
 50 D5 012A 240 TSTL R0 : ANY MORE DIGITS TO CONVERT?
 02 13 012C 241 BEQL 30\$: IF EQL NO
 EF 10 012E 242 BSBB 20\$: CONVERT NEXT DIGIT
 50 8ED0 0130 243 30\$: POPL R0 : RETRIEVE NEXT CHARACTER
 FECA' 30 0133 244 BSBW DCLSPUTCHAR : PUT CHARACTER IN EXPANSION BUFFER
 05 0136 245 RSB :
 0137 246 :
 0137 247 : DSABL LSB
 0137 248 :

```

0137 250 .SBTTL EDIT THE STRING
0137 251 :+
0137 252 : DCL$EDIT - EDIT THE STRING
0137 253 :
0137 254 : THIS ROUTINE IS CALLED TO EDIT THE INPUT STRING. THE RESULTANT STRING
0137 255 : IS RETURNED VIA THE INPUT DESCRIPTOR.
0137 256 :
0137 257 : INPUTS:
0137 258 :
0137 259 : R0 = EDIT FLAGS
0137 260 : R1 = LENGTH OF INPUT STRING
0137 261 : R2 = ADDRESS OF INPUT STRING
0137 262 :
0137 263 : OUTPUTS:
0137 264 :
0137 265 : R1 = LENGTH OF EDITED STRING
0137 266 : R2 = ADDRESS OF EDITED STRING
0137 267 :-
0137 268 :
0137 269 DCL$TRIM::
50 18 D0 0137 270 MOVL #EDIT_M_UPCASE!EDIT_M_TRIM, R0 ; SET UPCASE AND TRIM FLAGS
08 11 013A 271 BRB DCL$EDIT
013C 272 :
013C 273 DCL$SQUEEZE::
50 14 D0 013C 274 MOVL #EDIT_M_UPCASE!EDIT_M_COLLAPSE, R0 ; SET UPCASE AND COLLAPSE FLAGS
03 11 013F 275 BRB DCL$EDIT
0141 276 :
0141 277 DCL$UPCASE::
50 10 D0 0141 278 MOVL #EDIT_M_UPCASE, R0 ; SET UPCASE FLAG
0144 279 :
0144 280 DCL$EDIT::
51 D5 0144 281 TSTL R1 : IS LENGTH ZERO
1C 13 0146 282 BEQL 6$ : YES, THEN DONE
50 D5 0148 283 TSTL R0 : ARE NO FLAGS SET
18 13 014A 284 BEQL 6$ : YES, THEN DONE
7E 53 7D 014C 285 MOVQ R3,-(SP) : SAVE R3,R4
53 52 D0 014F 286 MOVL R2,R3 : COPY ADDRESS OF STRING
54 52 D0 0152 287 MOVL R2,R4 : COPY ADDRESS OF STRING
7E 52 D0 0155 288 MOVL R2,-(SP) : COPY ADDRESS OF STRING
7E D4 0158 289 CLRL -(SP) : CLEAR FLAGS
015A 290 :
015A 291 :
015A 292 : IF CHARACTER IS A QUOTE THEN TOGGLE THE FLAG, AND COPY THE CHARACTER
015A 293 :
63 22 91 015A 294 $: CMPB #^A/'',,(R3) : IS CHARACTER A QUOTE
08 12 015D 295 BNEQ 7$ : NO, THEN BRANCH
6E 01 CC 015F 296 XORL #1,(SP) : YES, TOGGLE QUOTE FLAG
79 11 0162 297 BRB 30$ : COPY THE CHARACTER
0164 298 :
0098 31 0164 299 6$: BRW 50$ : ALL DONE
0167 300 :
0167 301 :
0167 302 : IF INSIDE QUOTES, THEN COPY THE CHARACTER
0167 303 :
73 6E E8 0167 304 ?$: BLBS (SP),30$ : IF IN QUOTES, COPY THE CHAR
016A 305 :
016A 306 :

```

05 50 00 E1 016A 307 : IF REMOVING COMMENTS, AND COMMENT CHARACTER WAS FOUND, THEN WE'RE ALL DONE.

63 63 61 7A 016A 308 :
016E 309 :
0171 310 :
0173 311 :
0173 312 :
0173 313 :
0173 314 : IF WE ARE COMPRESSING BLANKS, THEN EITHER SET THE BLANK FLAG, OR
0173 315 : REMOVE THE CHARACTER. ALSO, SAVE THE ADDRESS OF THE FIRST BLANK
0173 316 : IN CASE WE ARE TRIMMING BLANKS.

63 20 91 0173 317 10\$:
63 05 13 0176 318 :
63 09 91 0178 319 :
1E 12 0178 320 :
16 50 02 E0 017D 322 15\$:
04 50 03 E1 0181 323 :
0E 6E 02 E1 0185 324 :
07 50 01 E1 0189 325 16\$:
63 20 90 018D 326 :
03 6E 01 E2 0190 327 :
84 63 90 0194 328 17\$:
53 D6 0197 329 18\$:
4F 11 0199 330 :
019B 331 :
019B 332 :
019B 333 : IF WE ARE UPCASING THE STRING, THEN NOW DO SO.

1F 50 04 E1 019B 334 :
61 8F 63 91 019F 335 20\$:
38 1F 01A3 336 :
7A 8F 63 91 01A5 337 :
OC 18 01A9 338 :
E0 8F 63 91 01AB 339 :
2C 1F 01AF 340 :
FE 8F 63 91 01B1 341 :
26 1A 01B5 342 :
63 20 8A 01B7 343 :
21 11 01BA 344 25\$:
01BC 345 :
9C 11 01BC 346 :
01BE 347 29\$:
01BE 348 :
01BE 349 :
01BE 350 : IF WE ARE LOWERCASING THE STRING, THEN NOW DO SO.

1B 50 05 E1 01BE 351 :
41 8F 63 91 01C2 352 26\$:
15 1F 01C6 353 :
5A 8F 63 91 01C8 354 :
OC 18 01CC 355 :
C0 8F 63 91 01CE 356 :
09 1F 01D2 357 :
DE 8F 63 91 01D4 358 :
03 1A 01D8 359 :
63 20 88 01DA 360 :
01DD 361 27\$:
01DD 362 :
01DD 363 :
BBC #EDIT V UNCOMMENT,R0,10\$
CMPB #^A/!7,TR3)
BEQL 42\$
BBC #^X20,(R3)
BEQL 15\$
CMPB #^X09,(R3)
BNEQ 20\$
BBS #EDIT V COLLAPSE,R0,18\$
BBC #EDIT V TRIM,R0,16\$
BBC #2,(SP),18\$
BBC #EDIT V COMPRESS,R0,17\$
MOVB #^X20,(R3)
BBSS #1,(SP),18\$
MOVB (R3),(R4)+
INCL R3
BRB 40\$
BBC #EDIT V UPCASE,R0,26\$
CMPB (R3),#^A/a/
BLSSU 30\$
CMPB (R3),#^A/z/
BLEQU 25\$
CMPB (R3),#^XEO
BLSSU 30\$
CMPB (R3),#^XFE
BGTRU 30\$
BICB #^X20,(R3)
BRB 30\$
BRB 5\$
BBC #EDIT V LOWERCASE,R0,30\$
CMPB (R3),#^A/A/
BLSSU 30\$
CMPB (R3),#^A/Z/
BLEQU 27\$
CMPB (R3),#^XCO
BLSSU 30\$
CMPB (R3),#^XDE
BGTRU 30\$
BISB #^X20,(R3)

: BRANCH IF NOT IGNORING COMMENTS
: IS CHARACTER A COMMENT DELIMITER
: YES, THEN DONE

: IS CHARACTER A BLANK
: YES, PROCESS IT
: IS CHARACTER A TAB
: NO, THEN BRANCH
: BRANCH IF COLLAPSING
: BRANCH IF NOT TRIMMING
: REMOVE LEADING BLANKS
: BRANCH IF NOT COMPRESSING
: CONVERT SPACE OR TAB TO SPACE
: IF NOT FIRST BLANK, THEN SKIP IT
: COPY THE CHARACTER
: POINT PAST THE BLANK
: LOOP

: BRANCH IF NOT UPCASING
: CHECK LOW LIMIT OF LOW RANGE
: BR IF FAILED
: CHECK HIGH LIMIT OF LOW RANGE
: BR IF VALID CHARACTER
: CHECK LOW LIMIT OF HIGH RANGE
: BR IF FAILED
: CHECK HIGH LIMIT OF HIGH RANGE
: BR IF FAILED
: UPCASE THE CHARACTER

: BRANCH IF NOT LOWERCASING
: CHECK LOW LIMIT OF LOW RANGE
: BR IF FAILED
: CHECK HIGH LIMIT OF LOW RANGE
: BR IF VALID CHARACTER
: CHECK LOW LIMIT OF HIGH RANGE
: BR IF FAILED
: CHECK HIGH LIMIT OF HIGH RANGE
: BR IF FAILED
: LOWERCASE THE CHARACTER

01DD 364 : COPY THE CHARACTER TO THE RESULT STRING. SUBTRACT ONE FROM THE INPUT LENGTH
01DD 365 : AND CONTINUE.
01DD 366 :
84 83 90 01DD 367 30\$: MOVB (R3)+(R4)+ : MOVE CHARACTER
6E 02 CA 01E0 368 BICL #2,(SP) : CLEAR BLANK FLAG
6E 04 C8 01E3 369 BISL #4,(SP) : SET FIRST NON-BLANK SEEN
04 AE 54 DO 01E6 370 MOVL R4,4(SP) : SAVE ADDR OF LAST NON-BLANK
CF 51 F5 01EA 371 40\$: SOBGTR R1,29\$: LOOP TILL DONE
01ED 372 :
01ED 373 :
01ED 374 : GET THE RESULT STRING. DO ANY TRIMMING THAT IS NECESSARY
01ED 375 :
51 54 52 C3 01ED 376 42\$: SUBL3 R2,R4,R1 : CALCULATE NEW LENGTH
53 8E 7D 01F1 377 MOVQ (SP)+,R3 : RESTORE STACK
04 50 03 E1 01F4 378 BBC #EDIT V TRIM,R0,45\$: BRANCH IF NOT TRIMMING
51 54 52 C3 01F8 379 SUBL3 R2,R4,RT : CALCULATE SIZE AFTER TRIM
53 8E 7D 01FC 380 45\$: MOVQ (SP)+,R3 : RESTORE REGISTERS
05 01FF 381 50\$: RSB
0200 382 :
0200 383 .END

CNVRDX	00000000	R	02
DCL\$CBTA_DEC	000000F9	RG	02
DCL\$CBTA_HEX	0000009A	RG	02
DCL\$CBTA_HEXSTR	000000A7	RG	02
DCL\$CBTA_OCT	000000F4	RG	02
DCL\$CNVASCBIN	00000016	RG	02
DCL\$CNVNOEDIT	00000024	RG	02
DCL\$EDIT	00000144	RG	02
DCL\$PUTCHAR	*****	X	02
DCL\$SQUEEZE	0000013C	RG	02
DCL\$TRIM	00000137	RG	02
DCL\$UPCASE	00000141	RG	02
DIGIT_TABLE	00000006	R	02
EDIT_B_FLAGS	00000000		
EDIT_M_COLLAPSE	= 00000004		
EDIT_M_TRIM	= 00000008		
EDIT_M_UPCASE	= 00000010		
EDIT_V_COLLAPSE	= 00000002		
EDIT_V_COMPRESS	= 00000001		
EDIT_V_LOWERCASE	= 00000005		
EDIT_V_TRIM	= 00000003		
EDIT_V_UNCOMMENT	= 00000000		
EDIT_V_UPCASE	= 00000004		
RADIX	00000003	R	02
WRK_B_CMDOPT	FFFFFC3		
WRK_B_MAXPARM	FFFFFD0		
WRK_B_MINPARM	FFFFFD1		
WRK_B_PARMCNT	FFFFFCF		
WRK_B_PARMSUM	FFFFFC5		
WRK_B_RECALLCNT	FFFFFC4		
WRK_B_VALLEV	FFFFFC2		
WRK_B_VERBTYP	FFFFF486		
WRK_C_LENGTH	FFFFF492		
WRK_G_BUFFER	FFFFF896		
WRK_G_INPBUF	FFFFF986		
WRK_G_RESULT	FFFFF486		
WRK_K_LENGTH	FFFFF48E		
WRK_L_CHARPTR	FFFFFE6		
WRK_L_DISALLOW	FFFFF9AE		
WRK_L_ERRORRTN	FFFFF486		
WRK_L_EXPANDPTR	FFFFFE2		
WRK_L_IMAGE	FFFFF48A		
WRK_L_MARKPTR	FFFFFD2		
WRK_L_PAROUT	FFFFF9A2		
WRK_L_PMPTRADDR	FFFFF9A6		
WRK_L_PROMPTRTN	FFFFFC6		
WRK_L_PROPTR	FFFFFCA		
WRK_L_QUABLK	FFFFF9AA		
WRK_L_READRTN	FFFFFEA		
WRK_L_RECALLPTR	FFFFFB6		
WRK_L_RSLEND	FFFFF8A		
WRK_L_RSLNXT	FFFFF88		
WRK_L_SAVAP	FFFFF9F		
WRK_L_SAVFP	FFFFFD4		
WRK_L_SAVSP	FFFFFD6		
WRK_L_SIGNALRTN	FFFFF982		
WRK_L_SPECRTN			

WRK_L_TAB_VEC	FFFFFDDE
WRK_L_VERB	FFFFFBDE
WRK_W_FLAGS	FFFFFF0
WRK_W_FLAGS2	FFFFFFF2
WRK_W_IMGCHAN	FFFFFFEE
WRK_W_PMPTLEN	FFFFF99E

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
ABS .	00000000 (0.)	00 (0.)	NOPIC USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
\$ABSS	FFFFFFFC (0.)	01 (1.)	NOPIC USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
DCLSZCODE	00000200 (512.)	02 (2.)	NOPIC USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	11	00:00:00.05	00:00:01.86
Command processing	87	00:00:00.62	00:00:08.80
Pass 1	161	00:00:04.00	00:00:13.69
Symbol table sort	0	00:00:00.36	00:00:00.78
Pass 2	68	00:00:01.05	00:00:03.77
Symbol table output	8	00:00:00.06	00:00:00.06
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	337	00:00:06.17	00:00:28.99

The working set limit was 900 pages.

20016 bytes (40 pages) of virtual memory were used to buffer the intermediate code.

There were 20 pages of symbol table space allocated to hold 267 non-local and 33 local symbols.

383 source lines were read in Pass 1, producing 14 object records in Pass 2.

16 pages of virtual memory were used to define 11 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
\$255\$DUA28:[SYSLIB]SYSBLDMLB.MLB;1	0
\$255\$DUA28:[DCL.OBJ]DCL.MLB;1	3
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	3
TOTALS (all libraries)	6

346 GETS were required to define 6 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LI\$\$(CONVERT/OBJ=OBJ\$\$(CONVERT MSRC\$\$(CONVERT/UPDATE=(ENH\$\$(CONVERT)+EXECMLS/LIB+LIBS:DCL/LIB+SYSSLIBRARY:SYSBLDMLB/LIB

0069 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

